Write-up: NoobLab CTF

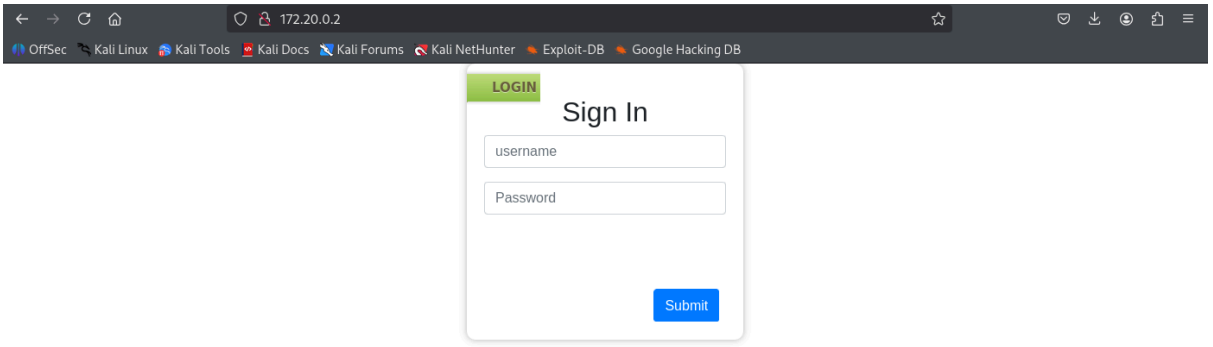
Autor(a): Patrícia
Data: 08 de Outubro de 2025
Alvo: NoobLab VM (172.20.0.2)
Origem: VulnHub
Link: https://www.vulnhub.com/entry/noob-1,746/

Resumo Executivo

Este relatório detalha a resolução da máquina de laboratório "NoobLab", um desafio de nível iniciante projetado para testar a metodologia e a persistência de um pentester. O acesso inicial foi obtido através de uma vulnerabilidade em servidor FTP, seguindo por uma complexa cadeia de enumeração e resolução de enigmas, esteganografia em imagens e decodificação de cifras simples, culminando no acesso SSH. A escalação de privilégios foi alcançada explorando uma configuração inadequada de permissões do sudo.

1. Reconhecimento (Reconnaissnace)

A fase inicial de reconhecimento foi efetuada, pela exploração da página exibida pelo IP da máquina.



A página possui uma tela de login, sendo testado inicialmente acessos óbvios, como admin/admin, root/root, admin/123456, admin/ , root/ , e espaço em branco para ambos, não obtendo sucesso.

O passo seguinte, foi a varredura de portas e serviços abertos pelo servidor da máquina. Para isso foi utilizado o NMap.

```
nmap -p- -sV -sC 172.20.0.2
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- -sV -sC -T4 -oN nmap_completo.txt 172.20.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-08 09:55 EDT
Nmap scan report for 172.20.0.2
Host is up (0.00083s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--    1 0        0              21 Sep 21  2021 cred.txt
|_-rw-r--r--    1 0        0              86 Jun 11  2021 welcome
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:172.20.0.1
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Login
55077/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 dc:e8:ad:80:35:81:c4:29:7e:cf:e4:70:f2:69:d9:96 (RSA)
|   256 46:20:20:03:9c:97:35:f6:2d:5d:62:4a:be:6c:95:8e (ECDSA)
|_  256 ae:90:88:f6:63:8d:dc:60:fa:ff:fc:70:12:e4:f4:1f (ED25519)
MAC Address: 00:0C:29:A6:6E:12 (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.13 seconds
```

Resultados Principais:

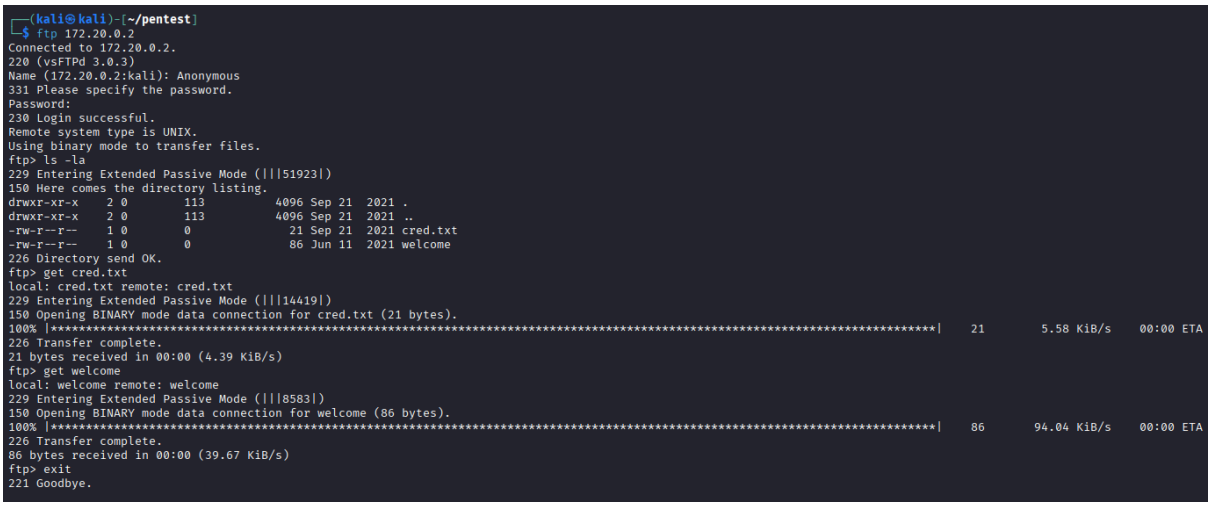
- **Porta 21/tcp:** FTP (vsftpd 3.0.3) com login anônimo permitido.
- **Porta 80/tcp:** HTTP (Apache httpd 2.4.29).
- **Porta 55077/tcp:** SSH (OpenSSH 7.6p1).

2. Enumeração de Serviços

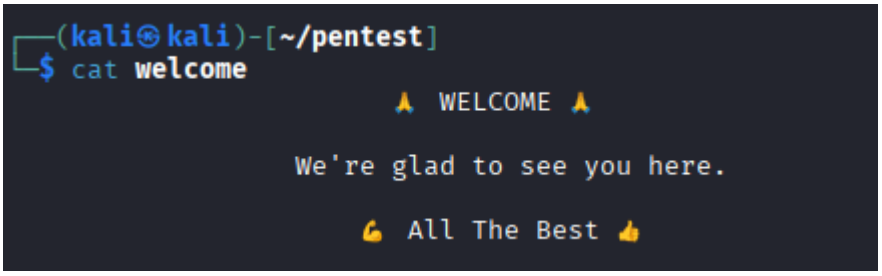
2.1. Análise do FTP (Porta 21)

Conforme observado através da varredura de portas, o acesso ao serviço ftp, foi realizado com o uso do usuário anônimo “**Anonymous**” permitido pelo serviço sem necessidade de senha.

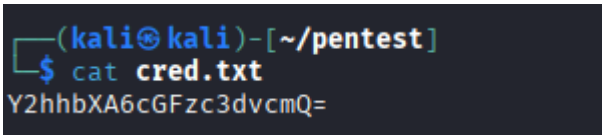
Após acesso concedido, realizou-se o download via “get” dos dois arquivos disponíveis para o usuário anônimo: **cred.txt** e **welcome**.



A análise do arquivo **welcome**, retornou uma mensagem de boas vindas, sem aparentes dados úteis para acessos.



A análise do conteúdo do arquivo **cred.txt,** retornou uma string codificada em Base64.



A string obtida pelo arquivo **cred.txt,** foi decodificada, retornando claros dados de usuário e senha:

- **Usuário:** champ
- **Senha:** password

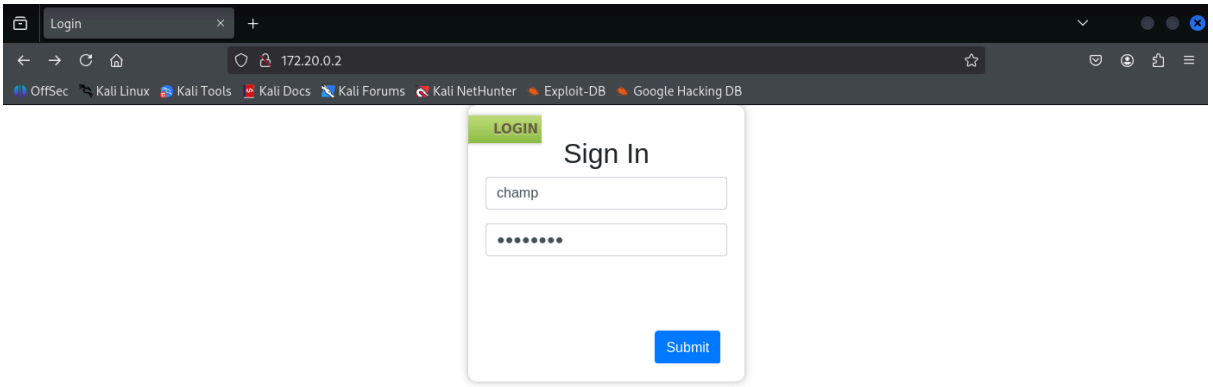
```
┌──(kali㉿kali)-[~/pentest]
└─$ echo "Y2hhbXA6cGFzc3dvcmQ=" | base64 -d > dados-cred.txt

┌──(kali㉿kali)-[~/pentest]
└─$ cat dados-cred.txt
champ:password
```

2.2. Análise de HTTP (Porta 80)

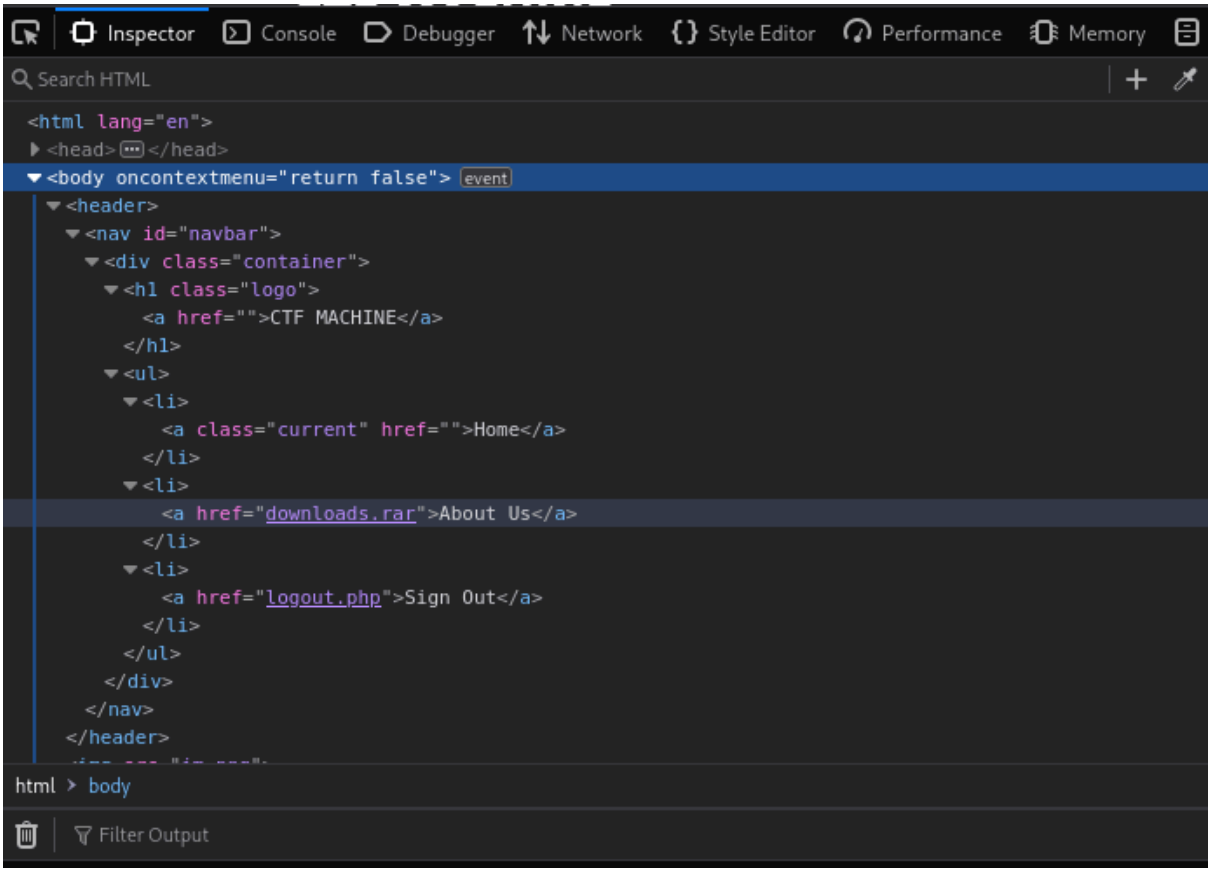
Conforme observado na fase de reconhecimento, existe uma página de login exibida através do serviço http para acesso da aplicação web.

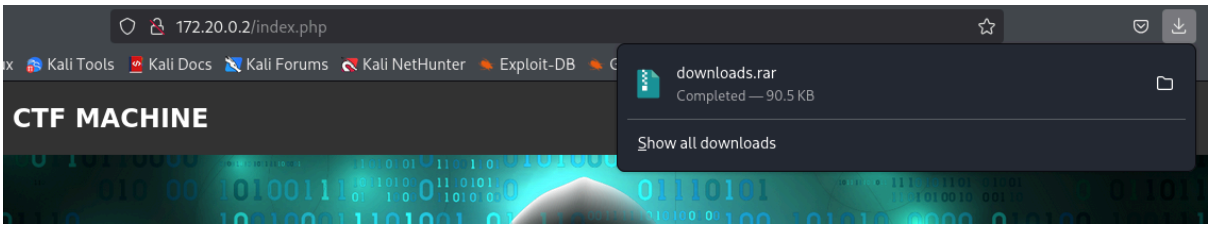
O usuário e senhas obtidas pela decodificação da string, **champ / password**, foram utilizadas como credenciais na tela de login, garantindo acesso a uma área autenticada da aplicação -> **index.php**.



3. Exploração e Obtenção das Pistas Finais

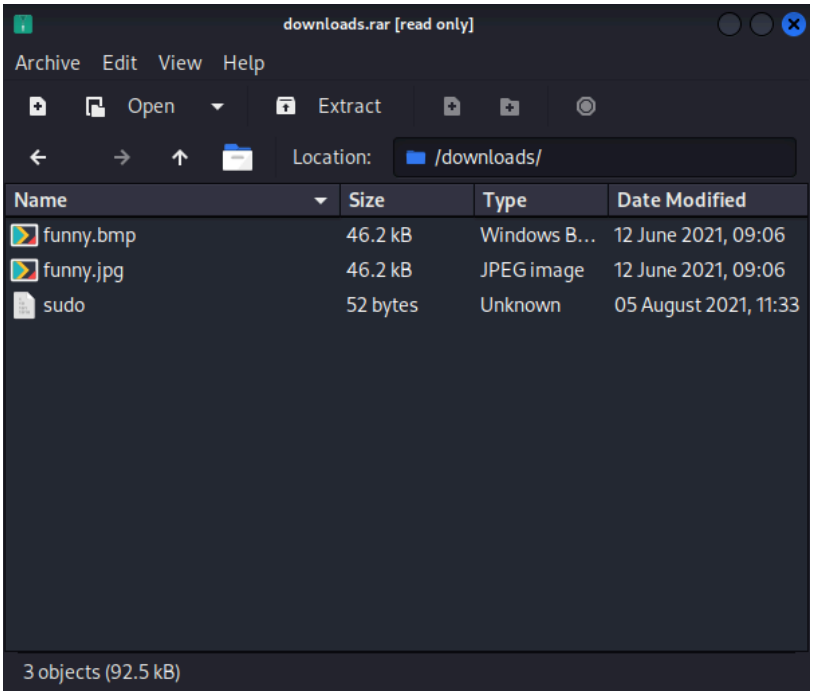
Ao explorar o código fonte da página, foi observado um link para um link "**About Us**" que disponibilizava o arquivo **downloads.rar** para download.





A extração deste arquivo revelou três itens cruciais:

- **funny.jpg (imagem)**
- **funny.bmp (imagem)**
- **sudo (arquivo de texto)**



3.1. Análise do Arquivo sudo

A análise do conteúdo do arquivo, revelou uma importante pista sobre o nome do arquivo.

```
┌──(kali㉿kali)-[~/pentest/NoobLab]
└─$ cat sudo
Did you notice the file name? Isn't is interesting?
```

3.2. Análise da Imagem Funny.jpg

A primeira imagem continha um arquivo oculto, extraído com o uso de steghide sem a necessidade de senha.

```
steghide extract -sf funny.jpg > jpg.txt
```

```
┌──(kali㉿kali)-[~/pentest/NoobLab]
└─$ steghide extract -sf funny.jpg > jpg.txt
Enter passphrase:
the file "hint.py" does already exist. overwrite ? (y/n) y
wrote extracted data to "hint.py".

┌──(kali㉿kali)-[~/pentest/NoobLab]
└─$ cat jpg.txt

┌──(kali㉿kali)-[~/pentest/NoobLab]
└─$ cat hint.py
This is_not a python file but you are revolving around.
well, try_ to rotate some words too.
```

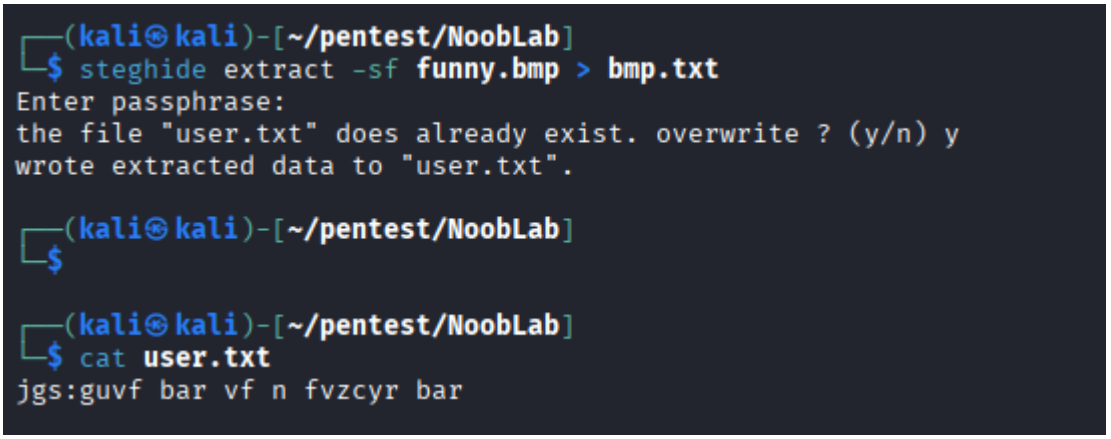
O conteúdo do **hint.py** era um enigma ("*rotate some words*"), apontando para a cifra **ROT13**.

3.3. Análise da Imagem Funny.bmp

A segunda imagem também continha dados ocultos, porém protegidos por senha. Nesse caso, foi utilizado o nome do arquivo **sudo** como senha, seguindo a pista dada pelo próprio arquivo, retornando da imagem, um arquivo intuitivamente chamado **user.txt**.

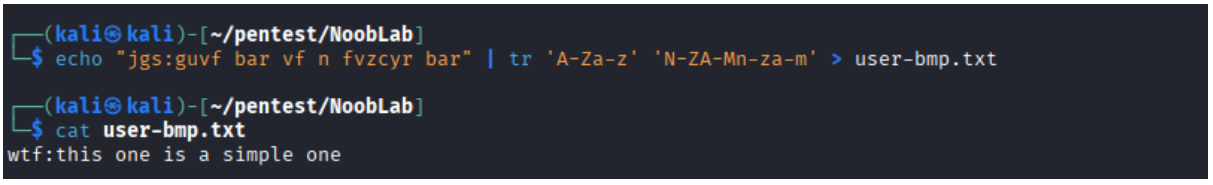
```
steghide extract -sf funny.bmp > bmp.txt
```

A análise do conteúdo do arquivo **user.txt,** revelou uma string evidentemente cifrada.



Seguindo a pista fornecida pelo arquivo **hint.py,** foi aplicada a decodificação da string por **ROT13**, retornando uma string de usuário e senha:

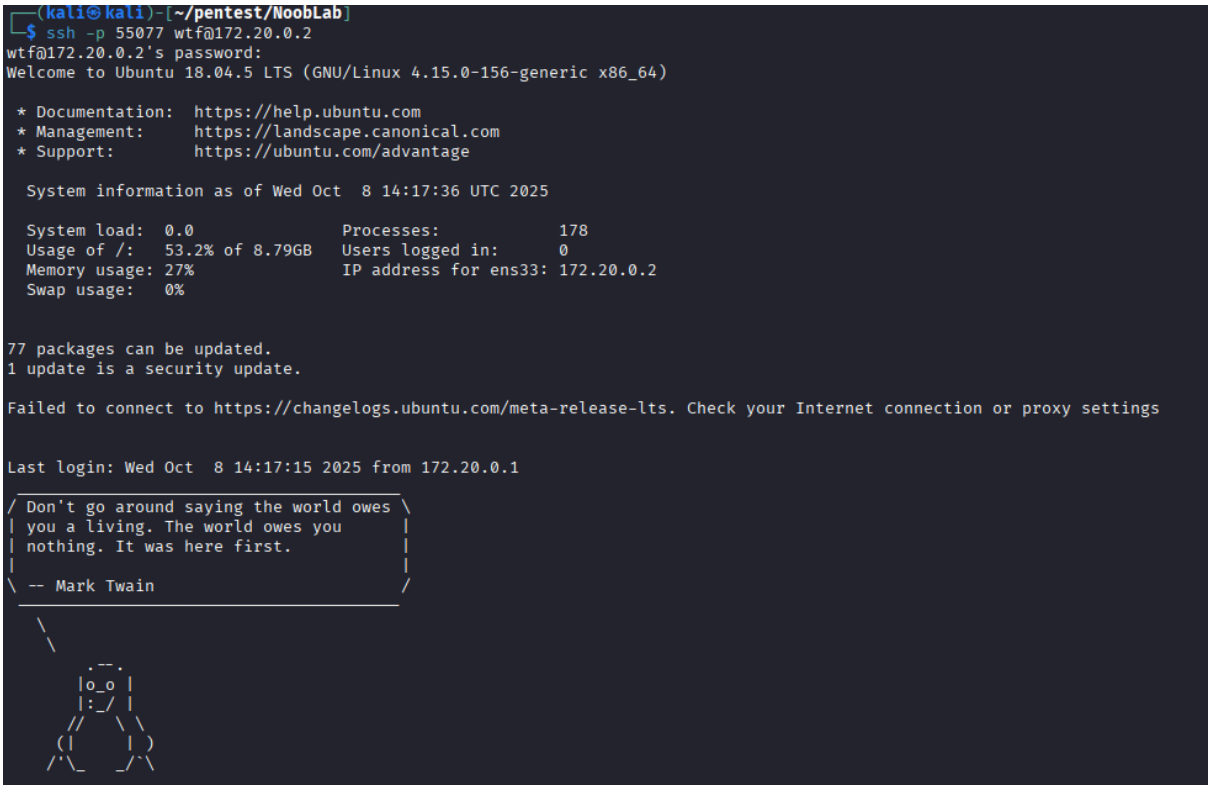
- **Usuário:** wtf
- **Senha:** this one is a simple one



4. Acesso Inicial

O acesso inicial, se constituiu, no uso do usuário e senha obtidos pelo arquivo **user.txt** no acesso SSH observado estar disponível na fase de reconhecimento.

```
ssh -p 55077 wtf@172.20.0.2
wtf@172.20.0.2's password: this one is a simple one
```



5. Escalação de Privilégios

Com o acesso SSH através do usuário **wtf**, foi realizado o comando para acesso ao usuário **root**, como teste de privilégios, utilizando a própria senha do usuário, escalando o acesso ao usuário **root** com sucesso.

```
sudo -i
[sudo] password for wtf: this one is a simple one
```

```
wtf@wtf:~$ sudo -i
[sudo] password for wtf:
root@wtf:~#
```

Conclusão

A máquina NoobLab, apesar de nomeada para iniciantes, exigiu uma metodologia rigorosa e a capacidade de correlacionar pistas de múltiplos vetores. A cadeia de exploração envolveu enumeração de serviços, análise de arquivos, quebra de enigmas de esteganografia e criptografia, e finalmente, a exploração de permissões inadequadas no sistema para obter acesso privilegiado.